

REMARKS

By this amendment, Claims 10, 11 and 20 have been amended and new Claims 21-49 added. Hence, Claims 1-49 are pending in the application. The specification has also been amended to correct minor typographical errors and deficiencies. It is respectfully submitted that the amendments to the claims and specification do not add any new matter to this application.

The Examiner is invited to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

If there are any additional charges, please charge them to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP



Edward A. Becker
Reg. No. 37,777

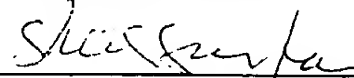
Dated: December 7, 2001

1600 Willow Street
San Jose, California 95125-5106
Telephone: (408) 414-1204
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231

on December 7, 2001 by


Sheila Severinghaus



AMENDED PORTIONS OF SPECIFICATION IN "MARKED UP" FORM

Text at Page 7, lines 8-11:

According to one feature, when the block of data is written to or read from nonvolatile memory a physical checksum verification procedure is performed on the block of data to determine whether the block of data was corrupted after performing the logical check on the data contained with the block of data. The physical checksum verification procedure may be performed in a storage device or another location, depending upon the requirements of a particular application.

Text at Page 11, Lines 16-24:

In one embodiment, prior to causing a block of data to be written to data storage unit 112, application 204 initiates a physical checksum calculation process 220 to generate a checksum value for inserting into the data block. In one embodiment, to generate the checksum value, a logical operation, such as an XOR or [AND] ADD operation is performed on the data that is contained in the data block. Based on the vector that results from performing the logical operation, and a desired checksum constant, a checksum value is selected and inserted into the data block. Thereafter, the integrity of the data block can be determined by performing the same logical operation on the data block and comparing the results with the checksum value stored with the data block.

CLAIMS:

- 1 1. (NOT AMENDED) A method for maintaining data integrity, comprising the computer-
2 implemented steps of:
3 generating checksum data by performing a physical checksum calculation on a block of
4 data in volatile memory;
5 after generating said checksum data,
6 performing a logical check on data contained within the block of data; and
7 if the block of data passes said logical check, then causing the block of data to be
8 written to nonvolatile memory.
- 1 2. (NOT AMENDED) The method of Claim 1 wherein the steps of generating checksum
2 data and performing a logical check are performed in response to a request to write said
3 block of data to nonvolatile memory.
- 1 3. (NOT AMENDED) The method of Claim 1 further comprising the step of writing the
2 checksum data to nonvolatile memory in association with writing said block of data to
3 nonvolatile memory.
- 1 4. (NOT AMENDED) The method as recited in Claim 3, further comprising the steps of:
2 after writing the block of data to nonvolatile memory,
3 causing the block of data and said checksum data to be read from nonvolatile
4 memory; and
5 performing a physical checksum verification procedure on said block of data
6 based on said checksum data, wherein the physical checksum verification
7 procedure indicates whether the block of data was corrupted subsequent to
8 performing the logical check on the data contained with the block of data.
- 1 5. (NOT AMENDED) The method as recited in Claim 1, further comprising the step of
2 performing one or more physical checksum verification procedures prior to writing the
3 block of data to nonvolatile memory, wherein the one or more physical checksum

4 verification procedures indicate whether the block of data was corrupted subsequent to
5 generating said checksum data.

1 6. (NOT AMENDED) The method as recited in Claim 1, wherein:

2 the step of performing a physical checksum calculation comprises the step of a software
3 application performing the physical checksum calculation on said block of data;
4 and

5 the step of performing a logical check on data contained with the block of data comprises
6 the step of said software application performing the logical check on data
7 contained with the block of data.

1 7. (NOT AMENDED) The method as recited in Claim 4, wherein:

2 the step of performing a physical checksum calculation comprises the step of an software
3 application performing the physical checksum calculation on said block of data;
4 the step of performing a logical check on data contained with the block of data comprises
5 the step of said software application performing the logical check on data
6 contained with the block of data; and

7 the step of performing a physical checksum verification procedure on said block of data
8 comprises the step of said software application performing the physical checksum
9 verification procedure on said block of data.

1 8. (NOT AMENDED) The method as recited in Claim 5, wherein:

2 the step of performing a physical checksum calculation comprises the step of an software
3 application performing the physical checksum calculation on said block of data;
4 the step of performing a logical check on data contained with the block of data comprises
5 the step of said software application performing the logical check on data
6 contained with the block of data; and

7 the step of performing one or more physical checksum verification procedures prior to
8 writing the block of data to nonvolatile memory comprises the step of one or more
9 components other than said software application performing the one or more
10 physical checksum verification procedures prior to writing the block of data to
11 nonvolatile memory.

1 9. (NOT AMENDED) The method as recited in Claim 4, further comprising the step of:
2 after performing the physical checksum verification procedure on said block of data,
3 storing the block of data as a backup version of the block of data, wherein the
4 backup version of the block of data is maintained separate from said block of data
5 in said nonvolatile memory.

1 10. (ONCE AMENDED) A method for storing data in a nonvolatile memory, comprising the
2 computer-implemented steps of:
3 determining a desired location in said nonvolatile memory for storing a data block;
4 inserting an address value in said data block, wherein the address value identifies the
5 desired location;
6 prior to performing an operation that stores the data block to the nonvolatile memory,
7 verifying that the address value contained within the data block correctly
8 identifies the location in nonvolatile memory into which the operation is going to
9 store the data block; and
10 performing the operation to store the data block to the nonvolatile memory only if the
11 address value contained within the data block correctly identifies the desired
12 location in nonvolatile memory into which the operation is going to store the data
13 block.

1 11. (ONCE AMENDED) The method as recited in Claim 10, further comprising the steps of:
2 after storing the block of data to nonvolatile memory,
3 reading the block of data from a location in nonvolatile memory; and
4 determining, based upon the address value contained within the data block, whether the
5 block of data was read from the desired location in the nonvolatile memory.
6 [comparing the address value contained within the data block with the location in
7 nonvolatile memory from which the data block was read.]

1 12. (NOT AMENDED) The method as recited in Claim 10, further comprising the step of
2 maintaining a mapping that identifies a specific location in said nonvolatile memory into
3 which said data block is to be stored.

1 13. (NOT AMENDED) The method as recited in Claim 10, wherein:

2 the step of determining a location in said nonvolatile memory comprises the step of
3 determining a plurality of locations in said nonvolatile memory for storing said
4 data block;

5 the step of inserting an address value in said data block, comprises the step of inserting a
6 plurality of address values in said data block, wherein the plurality of address
7 values identify multiple locations in said nonvolatile memory for which the data
8 block is to be stored; and

9 the step of storing the data block to nonvolatile memory comprises the step of storing the
10 data block in each of the multiple locations in nonvolatile memory only after
11 verifying the plurality of address values includes an address value that correctly
12 identifies the location in nonvolatile memory into which the data block is to be
13 stored.

1 14. (NOT AMENDED) A method for maintaining data integrity, comprising the computer-
2 implemented steps of:

3 performing a physical checksum calculation on a block of data;

4 after performing the physical checksum calculation,

5 performing a first physical checksum verification procedure on said block of data
6 prior to writing the block of data to nonvolatile memory, wherein the first
7 physical checksum verification procedure indicates whether the block of
8 data was corrupted subsequent to performing the physical checksum
9 calculation on the data contained with the block of data; and

10 if the block of data passes said first physical checksum verification procedure,
11 then causing the block of data to be written to nonvolatile memory.

1 15. (NOT AMENDED) The method as recited in Claim 14, further comprising the steps of:

2 after writing the block of data to nonvolatile memory,

3 causing the block of data to be read from nonvolatile memory; and

4 performing a second physical checksum verification procedure on said block of
5 data, wherein the second physical checksum verification procedure

6 indicates whether the block of data was corrupted subsequent to
7 performing the first physical checksum verification procedure on the data
8 contained with the block of data.

1 16. (NOT AMENDED) The method as recited in Claim 14, wherein the step of performing a
2 first physical checksum verification procedure includes the steps of performing a
3 plurality of physical checksum verification procedures on said block of data prior to
4 writing the block of data to nonvolatile memory, wherein the plurality of physical
5 checksum verification procedures indicate whether the block of data was corrupted
6 subsequent to performing the physical checksum calculation on the data contained with
7 the block of data.

1 17. (NOT AMENDED) The method as recited in Claim 14, wherein:
2 the step of performing a physical checksum calculation comprises the step of a software
3 application performing the physical checksum calculation on said block of data;
4 and
5 the step of performing a first physical checksum verification procedure on said block of
6 data comprises the step of a component other than said software application
7 performing said first physical checksum verification procedure on said block of
8 data prior to writing the block of data to nonvolatile memory.

1 18. (NOT AMENDED) The method as recited in Claim 17, wherein the step of performing a
2 physical checksum verification procedure on said block of data comprises the step of a
3 disk array component performing the physical checksum verification procedure on said
4 block of data, wherein the disk array component is configured to write the block of data
5 to disk only after verifying the integrity of the data block.

1 19. (NOT AMENDED) The method as recited in Claim 14, further comprising the step of:
2 after performing the physical checksum calculation,
3 performing a logical check on data contained with the block of data; and
4 if the block of data does not pass said logical check, then not writing the block of
5 data to nonvolatile memory.

1 20. (ONCE AMENDED) The method as recited in Claim 19, [further comprising the step:]
2 wherein:
3 the step of performing a physical checksum calculation comprises the step of an software
4 application performing the physical checksum calculation on said block of data;
5 the step of performing a logical check on data contained with the block of data comprises
6 the step of said software application performing the logical check on data
7 contained with the block of data; and
8 the step of performing a first physical checksum verification procedure prior to writing
9 the block of data to nonvolatile memory comprises the step of one or more
10 components other than said software application performing one or more physical
11 checksum verification procedures prior to writing the block of data to nonvolatile
12 memory.

1 21. (NEW) The method as recited in Claim 11, wherein the step of determining, based upon
2 the address value contained within the data block, whether the block of data was read
3 from the desired location in the nonvolatile memory includes comparing at least a portion
4 of the address value contained within the data block with data that indicates the location
5 in nonvolatile memory from which the data block was read.

1 22. (NEW) The method as recited in Claim 10, wherein:
2 the step of verifying that the address value contained within the data block
3 correctly identifies the location in nonvolatile memory into which
4 the operation is going to store the data block is performed by a
5 storage device, and
6 the step of performing the operation to store the data block to the
7 nonvolatile memory only if the address value contained within the
8 data block correctly identifies the desired location in nonvolatile
9 memory into which the operation is going to store the data block is
10 performed by the storage device.

1 23. (NEW) A computer-readable medium for maintaining data integrity, the computer-
2 readable medium carrying one or more sequences of one or more instructions which,

3 when executed by one or more processors, cause the one or more processors to perform
4 the steps of:
5 generating checksum data by performing a physical checksum calculation on a block of
6 data in volatile memory;
7 after generating said checksum data,
8 performing a logical check on data contained within the block of data; and
9 if the block of data passes said logical check, then causing the block of data to be
10 written to nonvolatile memory.

1 24. (NEW) The computer-readable medium of Claim 23 wherein the steps of generating
2 checksum data and performing a logical check are performed in response to a request to
3 write said block of data to nonvolatile memory.

1 25. (NEW) The computer-readable medium of Claim 23 further comprising one or more
2 sequences of additional instructions which, when executed by the one or more processors,
3 cause the one or more processors to perform the step of writing the checksum data to
4 nonvolatile memory in association with writing said block of data to nonvolatile memory.

1 26. (NEW) The computer-readable medium as recited in Claim 25, further comprising one or
2 more sequences of additional instructions which, when executed by the one or more
3 processors, cause the one or more processors to perform the steps of:
4 after writing the block of data to nonvolatile memory,
5 causing the block of data and said checksum data to be read from nonvolatile
6 memory; and
7 performing a physical checksum verification procedure on said block of data
8 based on said checksum data, wherein the physical checksum verification
9 procedure indicates whether the block of data was corrupted subsequent to
10 performing the logical check on the data contained with the block of data.

1 27. (NEW) The computer-readable medium as recited in Claim 23, further comprising one or
2 more sequences of additional instructions which, when executed by the one or more
3 processors, cause the one or more processors to perform the step of performing one or
4 more physical checksum verification procedures prior to writing the block of data to

5 nonvolatile memory, wherein the one or more physical checksum verification procedures
6 indicate whether the block of data was corrupted subsequent to generating said checksum
7 data.

1 28. (NEW) The computer-readable medium as recited in Claim 23, wherein:

2 the step of performing a physical checksum calculation comprises the step of a software
3 application performing the physical checksum calculation on said block of data;
4 and

5 the step of performing a logical check on data contained with the block of data comprises
6 the step of said software application performing the logical check on data
7 contained with the block of data.

1 29. (NEW) The computer-readable medium as recited in Claim 26, wherein:

2 the step of performing a physical checksum calculation comprises the step of an software
3 application performing the physical checksum calculation on said block of data;
4 the step of performing a logical check on data contained with the block of data comprises
5 the step of said software application performing the logical check on data
6 contained with the block of data; and

7 the step of performing a physical checksum verification procedure on said block of data
8 comprises the step of said software application performing the physical checksum
9 verification procedure on said block of data.

1 30. (NEW) The computer-readable medium as recited in Claim 27, wherein:

2 the step of performing a physical checksum calculation comprises the step of an software
3 application performing the physical checksum calculation on said block of data;
4 the step of performing a logical check on data contained with the block of data comprises
5 the step of said software application performing the logical check on data
6 contained with the block of data; and

7 the step of performing one or more physical checksum verification procedures prior to
8 writing the block of data to nonvolatile memory comprises the step of one or more
9 components other than said software application performing the one or more

10 physical checksum verification procedures prior to writing the block of data to
11 nonvolatile memory.

1 31. (NEW) The computer-readable medium as recited in Claim 26, further comprising one or
2 more sequences of additional instructions which, when executed by the one or more
3 processors, cause the one or more processors to perform the step of:
4 after performing the physical checksum verification procedure on said block of data,
5 storing the block of data as a backup version of the block of data, wherein the
6 backup version of the block of data is maintained separate from said block of data
7 in said nonvolatile memory.

1 32. (NEW) A computer-readable medium for storing data in a nonvolatile memory, the
2 computer-readable medium carrying one or more sequences of one or more instructions
3 which, when executed by one or more processors, cause the one or more processors to
4 perform the steps of:
5 determining a desired location in said nonvolatile memory for storing a data block;
6 inserting an address value in said data block, wherein the address value identifies the
7 desired location;
8 prior to performing an operation that stores the data block to the nonvolatile memory,
9 verifying that the address value contained within the data block correctly
10 identifies the location in nonvolatile memory into which the operation is going to
11 store the data block; and
12 performing the operation to store the data block to the nonvolatile memory only if the
13 address value contained within the data block correctly identifies the desired
14 location in nonvolatile memory into which the operation is going to store the data
15 block.

1 33. (NEW) The computer-readable medium as recited in Claim 32, further comprising one or
2 more sequences of additional instructions which, when executed by the one or more
3 processors, cause the one or more processors to perform the steps of:
4 after storing the block of data to nonvolatile memory,
5 reading the block of data from a location in nonvolatile memory; and

6 determining, based upon the address value contained within the data block, whether the
7 block of data was read from the desired location in the nonvolatile memory.

1 34. (NEW) The computer-readable medium as recited in Claim 32, further comprising one or
2 more sequences of additional instructions which, when executed by the one or more
3 processors, cause the one or more processors to perform the step of maintaining a
4 mapping that identifies a specific location in said nonvolatile memory into which said
5 data block is to be stored.

1 35. (NEW) The computer-readable medium as recited in Claim 32, wherein:
2 the step of determining a location in said nonvolatile memory comprises the step of
3 determining a plurality of locations in said nonvolatile memory for storing said
4 data block;
5 the step of inserting an address value in said data block, comprises the step of inserting a
6 plurality of address values in said data block, wherein the plurality of address
7 values identify multiple locations in said nonvolatile memory for which the data
8 block is to be stored; and
9 the step of storing the data block to nonvolatile memory comprises the step of storing the
10 data block in each of the multiple locations in nonvolatile memory only after
11 verifying the plurality of address values includes an address value that correctly
12 identifies the location in nonvolatile memory into which the data block is to be
13 stored.

1 36. (NEW) The computer-readable medium as recited in Claim 33, wherein the
2 step of determining, based upon the address value contained within the data
3 block, whether the block of data was read from the desired location in the
4 nonvolatile memory includes comparing at least a portion of the address
5 value contained within the data block with data that indicates the location in
6 nonvolatile memory from which the data block was read.

1 37. (NEW) The computer-readable medium as recited in Claim 32, wherein:
2 the step of verifying that the address value contained within the data block
3 correctly identifies the location in nonvolatile memory into which

4 the operation is going to store the data block is performed by a
 5 storage device, and
 6 the step of performing the operation to store the data block to the
 7 nonvolatile memory only if the address value contained within the
 8 data block correctly identifies the desired location in nonvolatile
 9 memory into which the operation is going to store the data block is
 10 performed by the storage device.

1 38. (NEW) A computer-readable medium for storing data in a nonvolatile memory, the
 2 computer-readable medium carrying one or more sequences of one or more instructions
 3 which, when executed by one or more processors, cause the one or more processors to
 4 perform the steps of:
 5 performing a physical checksum calculation on a block of data;
 6 after performing the physical checksum calculation,
 7 performing a first physical checksum verification procedure on said block of data
 8 prior to writing the block of data to nonvolatile memory, wherein the first
 9 physical checksum verification procedure indicates whether the block of
 10 data was corrupted subsequent to performing the physical checksum
 11 calculation on the data contained with the block of data; and
 12 if the block of data passes said first physical checksum verification procedure,
 13 then causing the block of data to be written to nonvolatile memory.

1 39. (NEW) The computer-readable medium as recited in Claim 38, further comprising one or
 2 more sequences of additional instructions which, when executed by the one or more
 3 processors, cause the one or more processors to perform the steps of:
 4 after writing the block of data to nonvolatile memory,
 5 causing the block of data to be read from nonvolatile memory; and
 6 performing a second physical checksum verification procedure on said block of
 7 data, wherein the second physical checksum verification procedure
 8 indicates whether the block of data was corrupted subsequent to
 9 performing the first physical checksum verification procedure on the data
 10 contained with the block of data.

1 40. (NEW) The computer-readable medium as recited in Claim 38, wherein the step of
2 performing a first physical checksum verification procedure includes the steps of
3 performing a plurality of physical checksum verification procedures on said block of data
4 prior to writing the block of data to nonvolatile memory, wherein the plurality of physical
5 checksum verification procedures indicate whether the block of data was corrupted
6 subsequent to performing the physical checksum calculation on the data contained with
7 the block of data.

1 41. (NEW) The computer-readable medium as recited in Claim 38, wherein:
2 the step of performing a physical checksum calculation comprises the step of a software
3 application performing the physical checksum calculation on said block of data;
4 and
5 the step of performing a first physical checksum verification procedure on said block of
6 data comprises the step of a component other than said software application
7 performing said first physical checksum verification procedure on said block of
8 data prior to writing the block of data to nonvolatile memory.

1 42. (NEW) The computer-readable medium as recited in Claim 41, wherein the step of
2 performing a physical checksum verification procedure on said block of data comprises
3 the step of a disk array component performing the physical checksum verification
4 procedure on said block of data, wherein the disk array component is configured to write
5 the block of data to disk only after verifying the integrity of the data block.

1 43. (NEW) The computer-readable medium as recited in Claim 38, further comprising one or
2 more sequences of additional instructions which, when executed by the one or more
3 processors, cause the one or more processors to perform the step of:
4 after performing the physical checksum calculation,
5 performing a logical check on data contained with the block of data; and
6 if the block of data does not pass said logical check, then not writing the block of
7 data to nonvolatile memory.

1 44. (NEW) The computer-readable medium as recited in Claim 43, wherein:
2 the step of performing a physical checksum calculation comprises the step of an software
3 application performing the physical checksum calculation on said block of data;
4 the step of performing a logical check on data contained with the block of data comprises
5 the step of said software application performing the logical check on data
6 contained with the block of data; and
7 the step of performing a first physical checksum verification procedure prior to writing
8 the block of data to nonvolatile memory comprises the step of one or more
9 components other than said software application performing one or more physical
10 checksum verification procedures prior to writing the block of data to nonvolatile
11 memory.

1 45. (NEW) A storage device comprising:
2 a storage medium; and
3 a storage mechanism communicatively coupled to the storage medium, the storage
4 mechanism being configured to:
5 perform a set of one or more verifications on a block of data prior to
6 allowing the block of data to be written to the storage medium,
7 wherein the set of one or more verifications includes a physical
8 checksum verification that indicates whether the block of data was
9 corrupted subsequent to a physical checksum calculation being
10 performed on the data contained with the block of data, and
11 if the block of data passes the set of verifications, then allow the block of
12 data to be written to the storage medium.

1 46. (NEW) The storage device as recited in Claim 45, wherein the set of verifications
2 includes a logical check on the data contained within the block of data.

1 47. (NEW) The storage device as recited in Claim 46, wherein the storage mechanism is
2 further configured to perform the logical check after the physical checksum verification.

1 48. (NEW) The storage device as recited in Claim 45, wherein the storage mechanism is
2 further configured to:
3 cause the block of data to be read from the storage medium, and
4 perform a second physical checksum verification on the block of data that indicates
5 whether the block of data was corrupted subsequent to the prior physical
6 checksum verification being performed on the block of data.

1 49. (NEW) A storage device comprising:
2 a storage medium; and
3 a storage mechanism communicatively coupled to the storage medium, the storage
4 mechanism being configured to:
5 perform a logical check on data contained in a block of data after a
6 physical checksum calculation has previously been performed on
7 the block of data, and
8 if the block of data passes the logical check, then allowing the block of
9 data to be written to the storage medium.